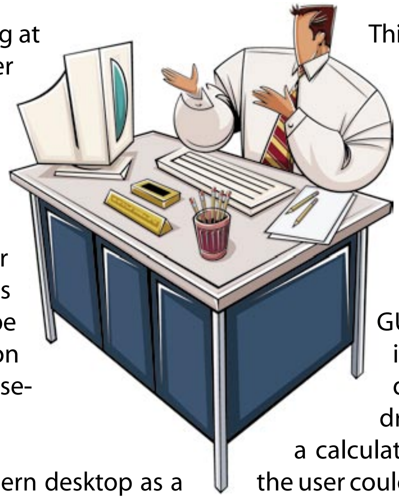
The Lazy Person's Guide to Controlling Technologies Part III: Mastering the Desktop

By Retired Air Force Major Dale J. Long

We spend hours every day staring at a window that has evolved over the last 30 years from a low resolution, text-based interface to a complex graphical interface. Though this dates me a bit, I can remember when going from 40-character to 80-character width on a monochrome monitor was a big deal. The computing environment has changed since then. Perhaps I should not describe the progression from text to graphics as evolution because that would imply survival of the most useful and appropriate features.



It might be more accurate to describe our modern desktop as a product of semi-intelligent design, a result of attempts at great innovation often impeded by technical compromises or ingrained bad habits. In this issue we will examine some of the quirks, idiosyncrasies and features that affect our desktop computing experience and examine why people often become frustrated with computers even when everything works as designed.

In The Beginning ...

There was text. And text with a command prompt was good enough for most of us, though learning arcane and often complex strings of text commands was a significant entry barrier to novice computer users. Predictably, there were people who wanted easier, more convenient ways to manipulate computer systems using pictures instead of words. The most famous early attempt at a graphical user interface (GUI) that most people are familiar with was the work done at Xerox Palo Alto Research Center (PARC). However, Xerox did not produce the first GUI. That distinction belongs to the U.S. Air Force Semi-Automatic Ground Environment (SAGE) project.

SAGE was a system developed by the North American Aerospace Defense Command beginning in the 1950s to track aircraft over North American airspace through real-time monitoring. SAGE computers collected text-based information and displayed it on a video monitor as moving icons. If this sounds familiar, it may be because SAGE technology was also used to develop air traffic control systems. Another early GUI was Sketchpad. Developed in 1963, it was the principal antecedent of modern computer-aided drafting programs. It was the first full GUI, using an x-y point plotter to display geometric forms on screen.

One other significant predecessor to Xerox PARC was the oNLine System (NLS) designed by Douglas Engelbart at the Stanford Research Institute (SRI) during the 1960s. The NLS was the first to employ hypertext linking a "mouse" (invented by Engelbart and Bill English) and framed windows.

This now brings us to Xerox PARC, where a team that included some SRI alumni drew upon these previous efforts and codified the windows, icons, menus and pointing devices (WIMP) system that became the foundation for the GUIs we are familiar with today.

Building on the Xerox PARC work, Apple Computer produced the first commercially successful GUI-based system: the Macintosh. Innovations included files on screen that looked like paper documents, file folder icons for directories and drop-down menus. The Macintosh also included a calculator, notepad and alarm clock applications that the user could place anywhere on the screen. The Mac's most unique feature was "drag and drop," which allowed users to easily move files and folders around with a mouse.

For some people Apple made deleting files entirely too easy. There was a trash can modification available that featured *Sesame Street's* Oscar the Grouch in a trash can. If you dropped in a file, Oscar would pop out and sing a line or two of his hit song, "I Love Trash." It was really cute.

Actually, it was entirely too cute. Speaking from personal experience, you should never mix file deletions, singing characters and an unsupervised three-year-old (who knows how to turn on the computer by himself) — unless you are prepared to find every file on your hard drive in the trash can awaiting deletion.

The Macintosh was followed several years later by a similar interface developed for IBM personal computers: Microsoft Windows. Despite (or perhaps because of) its lack of a trash can, Windows rapidly became the world's dominant desktop GUI. At that point, the GUI revolution ended.

GUI Stagnation

Try naming three things that have changed in GUI design in the last five years. Ten years? Fifteen years? For most people, the GUI invented by Apple and perpetuated by Microsoft is the only one they know. Mac users did experience some minor turbulence migrating from Classic to OS X, but it wasn't a big shift. If you are one of those rare people who use some flavor of Unix or Linux with Gnome or the Kopernicus Desktop Environment, you are still subject to the same defects inherent in commercial systems.

Defects? What defects? Let's start with visual attention. Every interaction with your computer's GUI requires your visual attention. On the old text-based command line systems, if you could touch type you only had to look at one area of the screen: the command prompt. In those days there were no distracting cut/

paste icons or function buttons cluttering up the screen. We had focus and we could keep it.

Unfortunately, the GUI on almost every computer I have worked on in the last 15 years requires you to divert your attention from what you are working on to find whatever graphical widget or button you need to activate. As an additional penalty, once you are done with the icon or menu you then have to relocate whatever you were typing or reading before you got the urge to reformat the text. The same penalty applies to scrolling, cutting, pasting, zooming or any other activity that requires refocusing your visual attention while you work.

Given the complexity of toolbars and all those tiny little icons all over the screen, it is no wonder people have trouble concentrating enough to produce any written work more complex than simple bulleted text. The loss of visual attention caused by the GUI does not let us hold our train of thought long enough to produce much more than one sentence at a time.

Navigation Hazards

Humans are capable of finding their way around by something called spatial navigation, which in general terms refers to how people learn to navigate through different environments using structured objects and landmarks. The longer you spend in a particular environment, the more familiar it becomes. Given enough time with a particular environment, what we learn through spatial navigation can eventually become reflex and muscle memory.

A good example of this is learning to drive a car. Beginning drivers usually fumble with things like finding the windshield wiper lever or overcorrecting on turns. Over time and with many repetitions, most will become somewhat proficient. When we drive a different car, most of us will fumble a bit with any controls that are different than the muscle memory we have developed, like trying to find the volume knob for the stereo or the fog light switch.

Now let's consider all the applications on your computer. Each time you use one it's like driving a different car. While the applications may have certain things in common, the functional equivalents of the stereo volume knob, wiper lever, etc., are all in slightly different places. Most people have a limit to how many different environments they can build up muscle memory for, so you may not spend enough time in any single application to become proficient.

Office Follies

I will now illustrate how software interferes with spatial navigation with a few examples from Microsoft Office. Please do not consider this Microsoft bashing. Most applications suffer from similar foibles, perhaps even more so because of attempts to differentiate them from what Microsoft produces. I use Office as an example because it is familiar and arguably the most dominant force on our desktops today. For reference, and if you would like to follow along, I am using MS Office 2000 for these examples.

First: menus. If I open MS Word and MS Excel and stack their windows so their menu bars are right on top of each other, I see that Microsoft did try to keep them consistent. However, at least on my screen, the Excel icon at the left side of the bar moves the Excel menu titles somewhat offline to the right of their counterparts in MS Word. Not a big difference, but enough that I have to change my focus to hit a menu item every time I use either program.

The main differences are within the drop-down menus. While the headings are similar, the internal choices are different. Granted this is probably because the programs do different things, but these differences are still an impediment to spatial navigation. The default menu preference compounds this complexity by only showing the menu items I have used most recently. While this is apparently an attempt to reduce complexity, it slows me down quite a bit if I need to find an item that is not immediately visible.

Once I have used a new item, my menu changes for the next few uses. If I do not use a function for a while it disappears from the menu, changing it again. Personally, I prefer having all the menu options visible all the time so it is always the same. *(You can turn off "Menus show recently used commands first" under Tools/Customize/Options.)*

Another difference between Word and Excel is how they deal differently with multiple file windows. If you open multiple documents in Word and later try to close one of them by clicking on the "X" button at the upper right corner of the screen, Word will only close that document. If you open multiple spreadsheets in Excel and then try closing any one spreadsheet with the same button, Excel will attempt to close all of them.

I have lost work on spreadsheets because I did not read the pop-up message asking if I wanted to save work on another sheet, clicked "No," and then watched helplessly as Excel closed sheets with unsaved work.

There are other things in Office, like differences in icon placement, size and interpretation that are not really showstoppers, but they can slow things down. Individually, they are barely noticeable and because we are used to them we do not normally notice the effect. Collectively, however, they can add up to a significant cognitive distraction.

There is a fine balance between consistency and choice. There are times I wish that operating system design were more totalitarian in how it governed human-computer interaction. If our GUI provided only one way to accomplish any given task, we would have an easier time learning to navigate our systems regardless of how we customized the arrangement of our desktops.

The closest personal example I can come up with is when I was stationed in England and had to learn to drive on the left side of the road with a car that had the steering wheel on the right side. It was unfamiliar and uncomfortable, and I went round a roundabout in Cambridge the wrong way (only once, though). But when I turned the steering wheel to the left, the car went left. The accelerator, brake and clutch were in the same relative positions as a car made for American drivers. Once I figured out which control lever turned on the lights and which one turned on the windshield wipers, muscle memory eventually took over.

GUI 2.0

Here are some GUI changes I would like to see.

Let's start with the menus. The first seven menus in MS Word, for example, have from 12 to 18 different line items, which are roughly twice as many options than the average human mind can handle at one time, if you believe cognitive scientists. The number of menu choices at any level should be reduced to five or six items. You will navigate faster with fewer menu options and have a better chance at building muscle memory.

Group things logically. In MS Word, why is the Tracking Changes function under the Tools menu instead of the Edit menu? Why are there both Customize and Options choices? Why is there an Insert menu and then an Insert sub-menu under Table?

Let users manage their own menus. One good example of a customizable application is the Microsoft Office 2000 Shortcut Bar. It is the only piece of screen real estate where I have relatively complete control within its configuration limits. It has my buttons, in my order, at the size and position I specify. I would really like the same control throughout the rest of the desktop. Finally, it just seems wrong that I have to shut down my computer by clicking the "Start" button.

The Paths of Least Resistance

Screen ergonomics could benefit from a redesign. There are four points on a computer monitor that anyone can hit with the cursor without looking: the corners. After decades of GUI research and development, you would think at least one operating system or major software application would use the screen corners for something other than controlling a screensaver. Granted, you can hide some toolbars and bring them up by moving the cursor to that edge of the screen on Windows and Mac, but that functionality has nothing to do with the actual work you are doing.

Many smart people have spent lots of time debating whether computer systems should be application-centered or document-centered. Everyone tends to work one way or the other. If you open the application first and then retrieve the document you want, you are application-centric. If you use your file manager to open the file you want, you are document-centric. Operating system designers do their best to accommodate both styles.

Yet, neither concept includes using the only spots on the desktop you can hit with a mouse without losing visual attention. How about a spatial-centric system? How tough would it be to make an operating system or application that can open a list of recently saved files; save the current file; get information on the current document; check e-mail; open the Control Panel; or open a task list and switch applications by moving the cursor to a screen corner?

Better yet, let us choose the functions we want the corners to perform and add function keys to save and close instead of just save files. We can be trusted with this functionality.

However, the root cause of our problems is that …

GUIs Are Designed For Beginners

System designers devoted considerable time and energy developing an interface that beginners could learn to use within an hour. Making it easy for new users is what drove the appeal of personal computers and made them the dominant information processing force in the world.

However, we are not beginners any more. We should be moving to more sophisticated interfaces. We perpetuate the evolutionary stagnation of our computing environment by recycling familiar defects and pretending it is "user-oriented" or "user-centric" design. If companies really want to be user-oriented, they should start weeding out the defects not keep building them in.

Closing Words

I do not believe small steps will work for evolving computer interfaces. What we need is another revolutionary change similar to moving from the command line to the GUI. This is what I think that means in practical terms. Let's say we see an image on a Web site we want to flip horizontally, shrink 50 percent and e-mail to a friend. Which of the following methods would you prefer?

Using our "modern" GUI: Right click on the image; save to file; open in a graphics program; use several mouse clicks to flip it; more mouse clicks to save it at 50 percent size reduction; open e-mail; attach the file; address the e-mail; and send.

Or right click on the image and say: "Computer, flip image horizontal, save image jpeg minus 50, e-mail saved image to Sills comma Dwight?"

The technology exists to do this now. So why don't we?

Until next time, Happy Networking!

Long is a retired Air Force communications officer who has written regularly for CHIPS since 1993. He holds a Master of Science degree in Information Resource Management from the Air Force Institute of Technology. He is currently serving as a telecommunications manager in the U.S. Department of Homeland Security. CHIPS

CHIPS Writing Guidelines

CHIPS welcomes articles from our readers. Please submit articles via e-mail as Microsoft Word or text file attachments to chips@navy.mil or by mail to Editor, CHIPS, SSC Charleston, 9456 Fourth Ave, Norfolk, VA 23511-2130. If submitting your article by mail, please send the article on disc with a printed copy. To discuss your article with a CHIPS editor, call (757) 444-8704 or DSN 564-8704.

CHIPS is published quarterly. Our deadline dates are: Feb. 1, April 1, Aug. 1 and Oct. 1. Go to our Web site at http://www.chips.navy.mil/chipsguidelines.html/ for more information. CHIPS